

PATENT

Attorney Docket
No. 10001-31297

APPLICATION FOR
UNITED STATES LETTERS PATENT
FOR
APPARATUS AND METHOD FOR
COORDINATING ACTIVITIES OF
ONE OR MORE COMPUTERS

SPECIFICATION

TO WHOM IT MAY CONCERN:

Be it known that Andrew J. Wardrop, a citizen of the United States of America, resident of Lakeville, Minnesota, has invented a certain new and useful invention of which the following is a specification:

CERTIFICATE OF EXPRESS MAIL

"Express Mail" mailing label number EH655098725US
Date of Deposit, OCTOBER 11, 2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" on the date indicated above and is addressed to the Assistant Commissioner For Patents, Washington, D.C. 20231

Lloyd Silveira

**APPARATUS AND METHOD
FOR COORDINATING ACTIVITIES
OF ONE OR MORE COMPUTERS**

5 CROSS-REFERENCE TO RELATED APPLICATION

This application claims benefit of Provisional Application No. 60/239,606 filed October 11, 2000 for "Method and Apparatus for Coordinating Activity in Multiple Computers Using GPS" by Andrew J. Wardrop.

10 BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to coordinating the activities of multiple computers. Specifically, the present invention relates to coordinating the activities of multiple computers using a common reference time point such as a Global Positioning System one pulse-per-second signal.

15 2. Description of the Related Art

Conventional computer systems typically use a central source for generating global interrupt signals. These signals typically are generated by a high quality, but imperfect, clock source. Use of such global interrupt signals by conventional computer systems requires a physical connection among the systems to coordinate activities using the global interrupt signals.

20 There exists a continuing need for a way to coordinate the activities of multiple computers that are not physically connected to one another using a precise reference time point.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for coordinating activity among multiple computers using a common reference time point such as a one pulse-per-second (“PPS”) signal available from the Global Positioning System (“GPS”) or generated locally. Based upon a common reference such as a PPS signal, interrupts can be generated at the same time within multiple computers despite each individual computer having its own imperfect clock. The interrupts can be generated within the computers at intervals shorter than the PPS signal. Multiple GPS receivers can each provide a PPS signal, and a consensus time point based upon the PPS signals can be used for coordinating the activities of the computers. A local PPS generator can be used by the computers if the GPS PPS signal becomes unavailable.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is an overall schematic diagram of the present invention;

FIG. 2 is a detailed schematic diagram of one computer according to the present invention; and

FIG. 3 is a schematic diagram of GPS timing alignment according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 1, an apparatus according to the present invention comprises GPS receiver 4150 and computer 10 having a mission timer 1000 that generates interrupts 1500, a reload register 2000 that transmits a reload value to mission timer 1000, timer capture

register 4100 that receives a PPS signal from GPS receiver 4150, and timer adjustment software 3000. The present invention may further comprise timer capture register 4200, GPS receiver 4250, timer capture register 4300, GPS receiver 4350, timer capture register 4400, and local PPS generator 4450.

5 1. Mission Timer

Referring to FIG. 1, computer 10 has a mission timer 1000. Mission timer 1000 generates interrupts 1500 that may be used to synchronize the internal operations of computer 10 with an external signal, such as a GPS PPS signal. Computers 20 and 30 also may include a mission timer (not shown). The mission timers (not shown) on computers 20 and 30 function in the same manner as mission timer 1000. Using such mission timers, the internal operations of computers 20 and 30, and any other associated computer, may be synchronized to one another.

Mission timer 1000 operates by counting down the clock signal from local oscillator 1100. Oscillator 1100 may operate at any appropriate clock rate. Referring now to FIG. 2, mission timer 1000 comprises a countdown timer 1200 and a rollover counter 1300. Countdown timer 1200 decrements with each clock tick generated by local oscillator 1100 and reloads a reload value from reload register 2000 when a count of zero is reached. Also, when a count of zero is reached by countdown timer 1200, a maskable interrupt 1500 is generated. Interrupt 1500 may be used to synchronize the internal operations of computer 10. Rollover counter 1300 counts the number of times that countdown timer 1200 counts down to zero in a given interval. In an alternative embodiment, countdown timer 1200 may use an incrementing timer rather than a decrementing timer.

When an independent reference time point is available, such as the GPS PPS signal, mission timer 1000 may be used to measure the frequency of local oscillator 1100 by counting the number of cycles of local oscillator 1100. When measuring the frequency of local oscillator 1100, mission timer 1000 uses a capture register (not shown) that reads a timer value at the time of a transition on an external signal such as a PPS signal.

2. Generating Reload Values

Mission timer 1000 generates interrupts 1500 based upon reload values transmitted from reload register 2000. Each time mission counter 1000 counts out (typically when zero is reached), mission timer 1000 loads a new reload value from reload register 2000. The following is a discussion of how such reload values are generated.

Generation of the reload values is based upon alignment with a PPS signal. While the present invention could distribute a PPS signal directly as a 1 Hz interrupt, the use of one or more of timer capture registers 4100, 4200, 4300, and 4400 and modification of reload values to align interrupts in multiple units to a scale shorter than the PPS signal is a preferred embodiment of the present invention. The use of timer capture registers 4100, 4200, 4300, and 4400 is discussed later.

In one embodiment of the present invention, timer adjustment software 3000 can be used to generate reload values at a desired frequency, typically between 10 Hz and 100 Hz. These reload values are transmitted from reload register 2000 to mission timer 1000. By generating reload values for a time period of 10 ms, an interrupt could be generated by mission timer 1000 every 10 ms for use in sequencing software activities. In terms often used by real time cyclic systems, a 1 second period would be called a major frame, and a 10 ms period would be called a minor frame. Accordingly, there would be 100 minor frames in a

major frame, numbered from 0 to 99 or 1 to 100, as shown in FIG. 3. It is preferable for the minor frame 100 to occur at the same time in the computers whose activities are being coordinated, such as computers 10, 20, and 30, so that interrupts will occur at the same time in the computers.

5 As an example, to generate a 100 Hz minor frame system when local oscillator 1100 has a nominal clock rate of 1 MHZ, mission timer 1000 uses nominal reload values of 10,000 (10 ms nominal). Referring to FIG. 3, when the PPS signal is asserted, the value of countdown timer 1200 (shown in FIG. 2) is captured. The point at which the PPS event is maintained can be anywhere within the major frame, but it is preferable that the point at which the PPS event is maintained is halfway through the last minor frame, for example, minor frame 100 shown in FIG. 3. In this example, the desired value of countdown timer 1200 when the PPS signal is asserted would be nominally 5,000 (*i.e.*, halfway through a 10,000 count minor frame).

10 At the beginning of the next minor frame (here, the beginning of minor frame 1), a reload register management function 3400 of software 3000 reads the value of
15 countdown timer 1200 captured when the PPS signal was asserted in minor frame 100. In this example, software 3000 computes the countdown timer 1200 error from the desired countdown timer 1200 value of 5,000. Based upon this error, reload register 2000 values for each minor frame are adjusted to bring the point at which the PPS signal is asserted to the
20 desired point at the end of the next major frame. For example, if a one second count of 1,000,027 was the last value used, and countdown timer 1200 had run long by 7 counts, a value of 1,000,020 would be used for the next second, divided up among the 100 minor frames. To achieve greater accuracy according to the present invention, when dividing errors

among the minor frames, round-off errors cannot be ignored, and round-off residuals must not accumulate.

Although using the right number of counts to predict the next point at which the PPS signal will be asserted will remove phase errors quickly, it is not mathematically stable. It is necessary to calculate the frequency of local oscillator 1100, and use the frequency of local oscillator 1200 as part of the determination of the count to be used for the next second. The correction applied adjusts both phase and frequency, but the frequency is adjusted on a longer time scale in order to provide a higher precision estimate of the local oscillator frequency.

The following is an example of an algorithm that could be used to correct phase errors:

$$\text{Count (N+1)} = \text{Frequency Estimate (N)} + A * (\text{Error(N)})$$

$$\text{Frequency Estimate (N+1)} = \text{Frequency Estimate (N)} + B * (\text{Error(N)} - \text{Error (N-1)})$$

Values of A = 1 to 1.25 and B = 0.1 appear to work reasonably well.

As would be known to those skilled in the art, other equations and other parameters derived from standard control theory may also be used instead of the above equations. Different equations and different parameters will affect the performance of the present invention, such as how quickly the system aligns with the PPS signal.

If the GPS PPS signal should become unavailable, mission timer 1000 will continue to generate interrupts based upon the reload values from reload register 2000. The reload values can be adjusted based on the last known local oscillator 1100 frequency, but the

system will eventually begin to drift. The frequency of local oscillator 1100 generally can be measured to an accuracy of $1/N$ ppm, where N is the time constant of the oscillator correction process. The clock will then start drifting at that rate, so the accuracy after A seconds will be A/B microseconds. As an example, if local oscillator 1100 is observed for 10 seconds, forming a 0.1 part-per-million measurement, after 100 seconds it may have drifted by 10 microseconds, just due to errors of measurement and extrapolation.

While aligning a minor frame boundary with the PPS signal could be done according to the present invention, it is preferable to maintain the PPS event within a minor frame, as described above. If the PPS event was maintained at the edge of a minor frame, half of the values of counter timer 1200 captured when the PPS signal is asserted would be in the next minor frame, after counter timer 1200 had received another reload value from reload register 2000. Because the present invention may use different reload values to coordinate the activities of multiple computers, the present invention would have to keep track of the reload values in order to calculate the error of local oscillator 1100 if the PPS event was maintained at the edge of a minor frame.

As described above, every time countdown timer 1200 counts out, it loads a new value from reload register 2000 for the next count down period. In a preferred embodiment of the present invention, to achieve the highest accuracy, the reload values generated by software 3000 may vary from one period to the next. This can be done in several ways.

The first way, and a preferred embodiment since it uses the least hardware, is for software 3000 to react to every minor interrupt and place a new reload value into reload register 2000 to be used the next time countdown timer 1200 rolls over.

A second way would be to have software 3000 generate a sequence of reload values once per second, and to load all of these reload values into a hardware table (not shown). Hardware would advance to the next entry in the table every time the countdown timer 1200 reached zero.

5 A third way would be for software 3000 to generate a fractional reload value once per second, and then use hardware (not shown) to adjust the reload value every time that counter timer 1200 rolls over. This fractional reload value would consist of upper bits that represent whole countdown timer 1200 ticks and lower bits that represent a fraction of one countdown timer 1200 tick. Likewise, countdown timer 1200 would consist of upper bits that get decremented and lower bits that do not get decremented. Hardware (not shown) would add the fractional reload value to countdown timer 1200 value when the decrementing portion of countdown timer 1200 reached 0. This addition would sometimes result in a carry bit propagating from the lower bits to the upper bits, such that countdown timer 1200 would count one more tick between roll overs than if there were no carry.

15 The following is an example of using fractional reload values if the desired countdown timer 1200 count was 12.4. On the first cycle, countdown timer 1200 would count for 12 ticks and then rollover. The residual value of countdown timer 1200 would be 0.4 when it rolled over. At the rollover, 12.4 would be added to countdown timer 1200, resulting in a value of 12.8. In the second period, countdown timer 1200 would again count
20 down for 12 ticks, leaving a residual of 0.8. After adding 12.4 again, countdown timer 1200 would contain 13.2, and this time countdown timer 1200 would count for 13 ticks, leaving a residual of 0.2. When continuously repeated, the average countdown timer 1200 count out

period would approximate the desired 12.4. The maximum error would be less than one timer tick.

3. Timer Capture Register

Computers whose activities are being coordinated according to the present invention such as computers 10, 20, and 30 utilize either a GPS PPS signal or an internally generated PPS signal to coordinate activities. GPS receiver 4150 is used by computer 10 to receive the GPS PPS signal for coordinating activities. Computer 10 may also use GPS receivers 4250 and 4350 to receive additional GPS signals. In addition, local PPS generator 4450 may be used by computer 10 when GPS signals become unavailable.

Timer capture registers are used according to the present invention to capture the value of countdown timer 1200 when a PPS signal is asserted. Referring to FIG. 2, timer capture register 4100 captures the value of countdown timer 1200 when a PPS signal is received from GPS receiver 4150. Timer capture register 4200 captures the value of countdown timer 1200 when a PPS signal is received from GPS receiver 4250. Timer capture register 4300 captures the value of countdown timer 1200 when a PPS signal is received from GPS receiver 4350. Timer capture register 4400 captures the value of countdown timer 1200 when a PPS signal is received from local PPS generator 4450.

Each computer whose activity is being coordinated typically will receive from one to three PPS synchronization signals from the GPS receivers. In addition, each computer whose activities are being coordinated typically will receive from at least one GPS receiver a serial message indicating the universal time ("UTC"). Each computer typically will slave the generation of interrupts from mission timer 1200 to the PPS signals, such that a major frame begins at the same absolute time in every computer. If some of the GPS PPS signals are

missing, the computers whose activities are being coordinated will use the signals that are present to generate interrupts. If the GPS serial message is missing, the computer will use information from the other computers to determine UTC.

The activities of the computers are coordinated by basing the interrupts generated by mission timer 1000 of each computer upon a PPS signal. By adjusting the reload values used by mission 1200 independently in each computer, the interrupts in each computer can be aligned, even though the countdown timer 1200 of each computer is driven by a different local oscillator 1100. As described above, the reload values used by mission timer 1200 are adjusted to compensate for any errors in local oscillator 1100.

When GPS receiver 4250 and GPS receiver 4350 are used in addition to GPS receiver 4150, the GPS PPS signal can be received in a fault tolerant manner. Software may be used to examine the values captured by timer capture registers 4100, 4200, and 4300 to determine a consensus value, such as an average, as a best estimate of true time. While each computer captures the PPS signals independently, if they all use the same algorithm for forming a consensus value, they will choose the same best estimate of true time. The process for determining a consensus value can include disallowing missing or obviously erroneous PPS signals. In this manner, the computers can remain synchronized even if all GPS signals are not available.

Local PPS generator 4450 can be included in the present invention to maintain synchronization among computers in the case that all GPS receivers are not working properly. Since local PPS generator 4450 is unlikely to be exactly at 1 pulse-per-second, the system time alignment will drift with respect to UTC. Using a typical short term drift of 1 part-per-million, it will take at least 100 seconds before a computer is off from UTC by 1% of a

typical minor frame (0.1 ms). However, since the signal is provided to all computers, the computers will remain synchronized with each other.

There are two preferred embodiments of local PPS generator 4450. In one embodiment, local PPS generator 4450 is associated with a GPS receiver, and only provides a signal when the GPS receiver is not receiving signals from the GPS satellites. Such a PPS generator would actively align its PPS signal with the GPS derived PPS signal when the GPS signal was available, such that switching from the GPS PPS to the locally generated PPS would not perturb the system. Even if there are several such PPS signals that are drifting slowly with respect to each other, as long as each computer used the same time consensus algorithm, the computers would remain in relative synchronization.

In a second embodiment, local PPS generator 4450 is independent of the GPS receivers. The locally generated PPS signal will be expected to drift slowly with respect to the GPS derived PPS signals due to inaccuracies in the clock oscillator of the local PPS generator. While the GPS signals are available, local PPS generator 4450 would be measured for time offset and actual period, using a separate timer capture register to compare to the GPS derived PPS signals. When the GPS signals became unavailable, the timing adjustment algorithm would incorporate this known offset in determining the timing error and subsequent correction. Information about the offset and the local PPS actual period could be exchanged among the several computers to allow finer adjustment, but all computers must end up with the same offset and actual period values in order to maintain synchronization.

Whereas the present invention has been described with respect to specific embodiments thereof, it will be understood that various changes and modifications will be

suggested to one skilled in the art and it is intended that the invention encompass such changes and modifications as fall within the scope of the appended claims.